# WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, June 30, 2005

| Hide? | Set Name | Query | Hit Count |
|---|---|---|---|
| | *DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ* | | |
| ☐ | L9 | 20020102 | 3 |
| ☐ | L8 | (client sync) same (server sync) | 12 |
| ☐ | L7 | l3 or l4 | 7 |
| ☐ | L6 | L2 and ( command element) | 0 |
| ☐ | L5 | L2 and (protocol command element) | 0 |
| ☐ | L4 | L2 and (data structure) | 7 |
| ☐ | L3 | L2 and (data element) | 4 |
| ☐ | L2 | 20020102 | 24 |
| ☐ | L1 | SyncML | 176 |

END OF SEARCH HISTORY

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L7: Entry 2 of 7 File: PGPB May 13, 2004

DOCUMENT-IDENTIFIER: US 20040093342 A1
TITLE: Universal data mapping system

Abstract Paragraph:
A universal mapping system, including apparatuses and methods, which is configurable through use of selectable configuration files to perform the bi-directional mapping and conversion of data between different data structures and formats. Each configuration file uniquely corresponds to a particular type of device which stores mappable data in a structure and/or format different than those of other types of devices. The universal mapping system is operable to perform mappings of data according to priority rules which are definable in the configuration files and which govern the order in which the mappings of data elements are performed. The universal mapping system is also adapted to maintain associations between data elements resulting from prior mappings. Additionally, the universal mapping system is operable to persist unmappable data from a source during a mapping in one direction and to return the unmappable data to that source during a subsequent mapping in the opposite direction.

Application Filing Date:
20011113

Summary of Invention Paragraph:
[0005] Today, SyncML has emerged and serves as the "standard" protocol for communicating and synchronizing personal information between computers and hand-held devices. SyncML provides common formats for the exchange of personal information management ("PIM") data between such computers and hand-held devices, including the "vCard" format for contact data and the "vCalendar" format for events (i.e., appointments or other scheduled events) and tasks. However, the synchronization software vendors are still faced with difficulties created by the multitude of different data formats used by server computers, standalone computers, and hand-held devices to store personal information. For instance, for each contact, a PIM database on a server computer may have two, thirty character-long fields for the storage of a contact's address, while a first hand-held device may have one, fifty character-long field for storage of the contact's address and a second hand-held device, of a different type than the first hand-held device, may have three, twenty character-long fields for storage of the contact's address. Also, each of the server computer and hand-held devices may allow the storage of certain characters and disallow the storage of other characters in a unique manner. Additionally, each of the hand-held devices may map their PIM data to/from the vCard format, and its properties, in different ways. As a consequence, to enable the communication and subsequent synchronization of PIM data between the server computer's PIM database and the hand-held devices' PIM data stores, the different numbers and formats of fields must be mapped and/or converted to/from the vCard format by a server computer based, at least, upon on the direction of the transfer of PIM data, on the number and formats of the fields for data storage employed by the server and hand-held devices, on the allowed/disallowed characters, and on the manner in which the different hand-held devices map their PIM data to/from the vCard format.

Summary of Invention Paragraph:
[0006] The majority of synchronization software vendors handle such requisite mapping and conversion of PIM data in their software by employing a set of software drivers which are specifically written and tailored to map and convert PIM data to/from the hand-held devices supported by their synchronization software. Because of the differences in the manners in which each type of hand-held device may store and, otherwise, handle PIM data, the synchronization software must, generally, include a software driver for each type of hand-held device. The development of such software drivers requires the analysis of the PIM data store structures of each type of hand-held device and the mappings between their data fields and the vCard and server PIM data structures, the design of appropriate driver software, the implementation (i.e., programming or coding) of the driver software, and the testing of the driver software. Hence, the development of such software drivers requires the expenditure of substantial resources in terms of time and money and may be fraught with potential errors.

Summary of Invention Paragraph:
[0008] Broadly described, the present invention comprises a universal mapping system, including apparatuses and methods, which is configurable through the use of selectable configuration files to perform the bi-directional mapping and conversion of data between different data structures and formats. Each configuration file corresponds, in a one-to-one relationship, to a type of device which stores mappable data in a structure and/or format different than those of other types of devices. The universal mapping system is operable to perform mappings of data in accordance with priority rules which are definable in the configuration files and which govern the order in which the mappings of data elements (i.e., properties and/or fields) are performed. The universal mapping system is also adapted to maintain, in subsequent mappings, associations between data elements resulting from prior mappings. Additionally, the universal mapping system is operable to persist unmappable data from a source during a mapping in one direction and to return the unmappable data to that source during a subsequent mapping in the opposite direction.

Summary of Invention Paragraph:
[0009] More narrowly described, the present invention comprises a universal mapping system, including apparatuses and methods, which may be utilized, in one form, as a portion of a synchronization system for the synchronization of personal information management data between a server computer system and a plurality of client devices of different types. The universal mapping system is configurable to enable the bi-directional mapping and conversion of personal information management data which is stored on the server computer system in a first structure and/or format and on the client devices in different structures and/or formats which are unique to each particular type of client device. Preferably, the personal information management data includes contact information which is exchanged between the server computer system and the client devices in accordance with the SyncML protocol and the vCard format. Because each type of client device, generally, maps data to/from the vCard format in a different manner, the universal mapping system includes a configuration file for each type of supported client device. Each configuration file may be identified and selected for use, during a synchronization session, based on the type of client device which is exchanging contact information with the server computer system. Generally, each configuration file includes information which identifies mappings and/or priority rules for mapping data between the vCard format and the data structure and formats of a database of contact information of the server computer system. Also, each configuration file may include information which defines characters that may be allowed or disallowed on a corresponding type of client device, supply substitute characters to be used in place of the disallowed characters, defines maximum field sizes, and provides configuration values relevant to the mapping and/or conversion of data.

Summary of Invention Paragraph:

[0013] Importantly, the universal mapping system also provides the ability to maintain data mappings from previous synchronization and/or mapping sessions, thereby preserving the integrity of a client device's data. The universal mapping system, additionally, enables the beneficial establishment and use of priorities that govern data mapping between data structures, and the persistence of data which significantly minimizes the potential loss of a client device's data that might result because certain client device data may not be mappable. In addition, the universal mapping system advantageously provides for the configuration and implementation of size limits on mapped data to diminish the possibility of a synchronization server computer system sending mapped data to a client device which does not fit, and perhaps exceeds, the limited storage capacity for such data at the client device. Further, the universal mapping system enables the configuration and implementation of lists of characters that are not allowed for storage in a client device and enables the specification of substitute characters therefor, thereby avoiding the sending of mapped data including unacceptable characters to a client device.

Summary of Invention Paragraph:
[0014] Accordingly, it is an object of the present invention to enable the mapping of data between two different data structures and/or formats without expenditure of the resources associated with the development, programming, testing, and implementation of software drivers.

Summary of Invention Paragraph:
[0015] Another object of the present invention is to map data between two different data structures using a configuration file to specify information which controls the mapping process.

Summary of Invention Paragraph:
[0021] Still another object of the present invention is to enable the mapping of data between a common data structure and/or format and a proprietary data structure and/or format.

Summary of Invention Paragraph:
[0022] Still another object of the present invention is to enable the mapping of personal information management data between the vCard data structure and/or format and a proprietary data structure and/or format of a synchronization server database.

Detail Description Paragraph:
[0040] The client devices 104, preferably, comprise Internet-enabled wireless devices having personal information manager applications software which are operable to communicate dPIM data to the synchronization server system 102 and to receive iPIM data from the synchronization server system 102 in the vCard format and through use of the SyncML protocol. Exemplary client devices 104 include, but are not limited to, appropriately configured Internet-enabled wireless telephones, personal digital assistants, and other similar devices available now or in the future. The telecommunication network 106, preferably, includes the Internet and the appropriate facilities thereof. Communication link 108, preferably, comprises a wired bi-directional communication path having multiple channels to enable bi-directional communication between the synchronization server system 102 and the telecommunication network 106. Communication links 110, preferably, include wireless bi-directional communication paths which enable bi-directional communication between respective client devices 104 and the telecommunication network 106. It should be understood that while the present invention is described in relation to system architecture 100, the scope of the present invention is not limited to wireless client devices 104, to the Internet telecommunication network 104, to wired communication links 108, or to wireless communication links 110, but instead comprises all wired or wireless client devices 104 (including, for example, desktop and mobile personal computers) capable of managing and communicating PIM

data in any format and according to any protocol, all telecommunication networks 106 and the infrastructure and facilities thereof, and all wired or wireless communication paths 108, 110 and the facilities thereof.

Detail Description Paragraph:
[0047] Each entry 302 of the PIM specification configuration file 116, preferably, comprises a name element or tag ("<Nam>") 304 which specifies the name of the PIM specification file 118 to be utilized by a universal mapping program 126 to configure itself, as described below, for the mapping of contact information communicated between an associated type of client device 104 and the synchronization server system 102 according to the SyncML protocol and the vCard specification. As a consequence, once an entry 302 is identified by the synchronization engine program 122 as corresponding to the device type of the client device 104, the name of the PIM specification file 118 is also identified, thereby enabling selection of the appropriate PIM specification file 118 from the plurality of available PIM specification files 118. Each entry 302 also, generally, comprises: a manufacturer element or tag ("<Man>") 306 which identifies the name of the manufacturer of the associated type of client device 104, and a model number element or tag ("<Mod>") 308 which identifies the model number assigned to the associated type of client device 104 by the device's manufacturer. Each entry 302 may additionally contain other elements or tags, including for example and not limitation: hardware, software and version elements or tags ("<Hwv>", "<Swv>", and "<Fwv>") 310, 312, 314 which respectively identify the version of the hardware, software, and firmware present in and being utilized by the associated type of client device 104; an original equipment manufacturer element or tag ("<Oem>") 316 which identifies the name of the original equipment manufacturer of the associated type of client device 104 (i.e., since the associated type of client device 104 may have been produced by one manufacturer, but marketed under a different manufacturer's name); a user agent element or tag ("<Usr>") 318 which identifies the user agent utilized by the associated type of client device 104; and, a default element or tag ("<Def>") 320 which identifies whether the associated entry 302 qualifies as a default entry 302 for use by the universal mapping program 126 in the event that an entry 302 cannot, otherwise, be identified for a particular client device 104.

Detail Description Paragraph:
[0055] The software and data platform 112 of the synchronization server system 102 additionally comprises a SyncML server program 124 (i.e., sometimes also referred to herein as "SyncML server 124") having computer software instructions which, when executed by a processor of the system 102, is operable to call the universal mapping program methods MapToUC ( ) and MapTovCard ( ) upon receipt of a vCard or a request for an update of the dPIM data of a client device 104 received from the web server 120 and the sync engine 122, and to thereby initiate mapping of PIM data between the server iPIM storage object 114 and a vCard. The SyncML server 124 is also operable to handle protocol conversion and/or related tasks associated with the communication of vCards via the SyncML protocol. The software and data platform 112 further comprises a universal mapping program 126, described in more detail below, which enables the bi-directional mapping of PIM data between a vCard and the master iPIM storage object 114. Together, the universal mapping program 126 and the plurality of PIM specification files 118 form the universal mapping system 140.

Detail Description Paragraph:
[0056] In accordance with the preferred embodiment of the present invention, each client device 104, preferably, comprises a hardware platform (not shown) having a processor for executing software program instructions and manipulating data, one or more memories for storing software program instructions and data, one or more user interfaces (i.e., keypad, display, microphone, speaker), and a communication interface which enables the bi-directional communication of data (i.e., in the form of vCards) with the telecommunication network 106 via a communication link 110 (i.e., and, ultimately, with the synchronization server system 102 via

communication link 108). Each client device 104 also, preferably, comprises a software and data platform 128 including a client dPIM storage object 130 for storing dPIM data associated with contacts of the device's user, and a plurality of software programs having software instructions which cause the client device 104 to perform certain tasks and provide certain functionality when executed by the device's processor. The plurality of software programs comprises a personal information manager program 132 (i.e., sometimes also referred to herein as "PIM manager 132") for enabling the device's user to locally input, delete, edit, and view dPIM data, to initiate PIM data synchronization with the synchronization server system 102, and to store and retrieve dPIM data in and from the client dPIM storage object 130. The plurality of software programs also comprises a SyncML client software program 134 (i.e., sometimes also referred to herein as "SyncML client 134") and a vCard mapping software program 136 (i.e., sometimes also referred to herein as "vCard mapper 136") which, preferably, work in conjunction with the PIM manager 132 to synchronize PIM data in the client dPIM storage object 130 and the server iPIM storage object 114. The SyncML client 134 and vCard mapper 136 are, respectively, operable to enable the bi-directional communication of vCards with the telecommunication network 106 (i.e., and, ultimately, with the synchronization server system 102) via the SyncML protocol, and to bi-directionally map PIM data between a vCard and the client dPIM storage object 130.

Detail Description Paragraph:
[0057] FIG. 2 displays a block diagram representation of the universal mapping system 140 according to the preferred embodiment of the present invention. The universal mapper 126, described briefly above, is a component of the synchronization server system 102 that is responsible for the bi-directional mapping of contact information stored in a dPIM storage object 130 of a client device 104 and contact information stored in the iPIM storage object 114 of the synchronization server system 102, with the contact information of both storage objects 114, 130 being exchanged, preferably, in the form of vCards communicated therebetween via the SyncML protocol. The universal mapper 126 is designed as a framework from which standard vCard mapping objects are configurable to perform mappings specific to a type of client device 104 and to which custom mapping objects may be added at various times. The client specific mappings are defined by a PIM specification file 118, described above, that is identified and communicated (i.e., by name) to the universal mapper 126 by the sync engine 122. The PIM specification file 118 provides the universal mapper 126 with configuration elements which describe properties for the mapping objects, in addition to unique mappings of vCard properties and parameters to fields of the UC records of the server iPIM storage object 114.

Detail Description Paragraph:
[0060] In operation, the synchronization server system 102 and its various component programs function in accordance with methods of the preferred embodiment of the present invention described herein. As illustrated in the flowchart representation of a synchronization method 600 displayed in FIG. 6, the synchronization server system 102 initiates a synchronization session, at step 602, when the server 102 receives a vCard from a client device 104 that has constructed the vCard from its dPIM data (i.e., through appropriate operation of the client device's PIM manager 132 and vCard mapper 136) and has submitted the vCard to the synchronization server system 102 through operation of the client device's SyncML client 134 and communication of the vCard to the server 102 via telecommunication network 106 and communication paths 108, 110. Upon receipt of the vCard via the server's web server 120, the vCard is communicated internally to the server's sync engine 122, SyncML server 124, and universal mapper 126. Initiation of a synchronization session may also occur as a result of a request for updating of a client device 104 being made. Regardless of the event causing initiation of the synchronization session and the direction of the synchronization, operation of the synchronization server system 102 is substantially the same.

Detail Description Paragraph:
[0062] Once the sync engine 122 has identified the name of the PIM specification file 118 corresponding to the client device type, the SyncML server 124 initiates mapping operations by calling the appropriate universal mapper method, MapToUC (i.e., to map vCard properties to fields of a UC record) or MapToVCard (i.e., to map fields of a UC record to vCard properties), and by passing the name of the previously identified PIM specification file 118 to the universal mapper 126. In response, the universal mapper 126, at step 606, configures itself for mapping operations, in the direction appropriate for the method called by the SyncML server 124, according to a configuration method 800 displayed in FIG. 8 and described below. Then, at step 608, the synchronization method 600 branches to cause operation of the synchronization server system 102 according to the direction of mapping. If PIM data is to be mapped from a vCard to a UC record (i.e., the MapToUC method was called by the SyncML server 124), the universal mapping program 126, at step 610, maps vCard properties and parameters to a UC record object having iPIM types and sub-types and passes the UC record object. A detailed description of the employed mapping method 900 is described below with reference to FIG. 9. Then, at step 612, the sync engine 122 updates the server's iPIM storage object 114 with the data from the UC record object and, at step 614, terminates synchronization operations. If, at step 608, PIM data is to be mapped from a UC record to a vCard (i.e., the MapToVCard method was called by the SyncML server 124), the universal mapper 126, at step 616, maps fields of a UC record having iPIM types and sub-types into vCard properties and parameters (see FIG. 9) and passes the vCard to the SyncML server 124 for communication of the vCard, at step 618, to the client device 104 using the SyncML protocol. The synchronization method 600 then terminates at step 620.

Detail Description Paragraph:
[0071] FIGS. 9A, 9B, and 9C depict a mapping method 900 of the preferred embodiment which is utilized and implemented by the universal mapper 126 to map PIM data in the direction (i.e., to/from vCard) associated with the universal mapper method, MapToUC or MapToVCard, called by the SyncML server 124 as described above. Broadly, the mapping method 900 comprises steps of calling each configured vCard property mapper 206 and allowing each vCard property mapper 206 to process the vCard property for which it is responsible. Then, for unmapped properties, the respective PIM data is stored as unmappable data and added to the history/association table. More specifically, after starting operation according to the mapping method 900 at step 902, the universal mapper 126 examines the list of vCard property mappers of the PIM specification object 208 and determines whether all of the configured vCard property mappers 206 have been called. If so, the universal mapper 126 branches to step 918 described below. If not, the universal mapper 126 utilizes the list and mapping information of the PIM specification object 208 to call, or retrieve, the next vCard property mapper 206 in the sequence identified by the list and the appropriate MapToXX method of the mapper 206 according to the direction of the synchronization session.

Detail Description Paragraph:
[0075] At step 916, the universal mapper 126 determines whether there are any more properties or fields to map. If so, the universal mapper 126 branches operation back to step 904 described above. If not, at step 918, the universal mapper 126 locates all unmapped properties and/or field(s), saves the associated PIM data in the history/association table, and updates the history/association table accordingly to account for and persist the unmapped PIM data and allow its return, to a client device 104, in a future synchronization session which updates the client device 104. Then, at step 920, the universal mapper 126 transforms the mapped data to the proper format for the direction of the synchronization session (i.e., places the mapped data into a UC record object or a vCard) and returns the resulting transformed and mapped data (i.e., resultant data) to the sync engine 122 and/or SyncML server 124, as appropriate, for updating of the server iPIM storage object 114 or communication to the client device 104. The universal mapper 126

terminates operation in accordance with the mapping method 900 at step 922.

Detail Description Paragraph:
[0079] It should be understood that the description of the preferred embodiment of the present invention is for descriptive or exemplary purposes only, and that the apparatuses and methods of the present invention may be utilized in a wired or wireless communications environment to map data between two data storage structures and/or formats. Also, the communications environment may include communication networks or communication connections other than the Internet and the use of protocols and specifications other than SyncML and vCard. Additionally, the present invention's apparatuses and methods may be employed in data management architectures other than client/server architectures. In addition, the apparatuses and methods of the present invention may be utilized to map any type of data other than personal information management data.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L7: Entry 7 of 7 File: PGPB Mar 21, 2002

DOCUMENT-IDENTIFIER: US 20020035609 A1
TITLE: Location bookmark system and method for creating and using location information

Application Filing Date:
20010820

Summary of Invention Paragraph:
[0042] ii) means for encoding data elements relating to said location;

Summary of Invention Paragraph:
[0043] iii) means for storing said data elements on a storage medium;

Summary of Invention Paragraph:
[0048] In another aspect of the invention there is provided a virtual location bookmark for use with a system as described above in which the data elements are adapted to contain data representations of:

Brief Description of Drawings Paragraph:
[0066] FIG. 3a is an example of data structure contained in a location bookmark in accordance with the invention;

Detail Description Paragraph:
[0099] LBKs so created are indexed by the user in accordance with a data structure that can be personalized to reflect his preferences. In the present embodiment of the invention, this is realized through the system that offers the user the possibility to create directories and sub-directories before selecting in which of these the LBKs created shall be inserted.

Detail Description Paragraph:
[0107] FIG. 5 illustrates the activity diagram associated with this system functionality. When a request to access LBKs is received by the system (500), it presents to the user the content of the root directory (501). The user then selects a directory (502) so as to ascertain its content: either one or more sub-directories or LBKs, which the user can then select. Where the user selects a sub-directory for visualization (502), its content is presented to him by the system. The user may repeat this operation so as to browse through the data structure until he finds the desired LBK. Where the user selects a LBK (506), its details are presented to the user on the device's screen (507).

Detail Description Paragraph:
[0126] 5--By using protocols such as Bluetooth and IrDA, with one or more LBKs as an attachment, copies of LBKs can be beam from one user to other(s). In this case, copies of LBK are stored locally into the wireless device and can be easily synchronized with the user central database through protocol such as SyncML.

CLAIMS:

1. A system for the creation and management of bookmarks relating to a location

comprising: a) a data server comprising; i) processor means for processing data; ii) means for encoding data elements relating to said location; iii) means for storing said data elements on a storage medium; iv) means for selectively accessing said data; v) data transceiver means; b) at least one user device; c) a data communication network adapted to connect said user device to said data server.

2. A system as claimed in claim 1 in which the data elements are adapted to contain data representations of: a) the geographical position of the location; and b) an identifier associated with the location.

3. A system as claimed in claim 2 in which the geographical position data elements comprise: a) the latitude associated with the location; and b) the longitude associated with the location.

5. A system as claimed in claim 3 further comprising data elements which are adapted to contain data representations of the altitude associated with the location.

6. A system as claimed in claim 4 further comprising data elements which are adapted to contain data representations of the identification of the author of the bookmark.

7. A system as claimed in claim 6 further comprising data elements which are adapted to contain data representations of the accuracy of the data representations of the latitude, the longitude and the altitude.

8. A system as claimed in claim 1 in which the data elements are adapted to contain data representations of: a) the latitude associated with the location; b) the longitude associated with the location; c) an identifier associated with the location; and d) the altitude associated with the location.

9. A virtual location bookmark for use with a system as described in claim 1 in which the data elements are adapted to contain data representations of: a) the geographical position of the location; and b) an identifier associated with the location.

10. A virtual bookmark as claimed in claim 10, wherein the geographical position data elements comprise; a) the latitude associated with the location; and b) the longitude associated with the location.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L7: Entry 3 of 7 File: PGPB Jan 2, 2003

DOCUMENT-IDENTIFIER: US 20030004937 A1
TITLE: Method and business process to maintain privacy in distributed recommendation systems

Application Filing Date:
20010913

Detail Description Paragraph:
[0117] By expressing the context in the service history log 110 in XML, the stored expression is both human and machine readable, it defines the content, and it defines the hierarchical structure of the content. XML also separates the appearance of the content from the structure of the content, so that the content can be displayed in any format by using customized style sheets in each different type of display device. Extensible Stylesheet Language (XSL) can provide flexible document presentation, enabling the content of an XML file to be displayed on the large display screen of a personal computer, as well as in the microbrowser 102. Messages exchanged between the wireless device 100 and the network server 140 can include XML files carried in the Simple Object Access Protocol (SOAP) messaging protocol or the SyncML synchronization protocol. For additional background on XML, see the book by Heather Williamson, XML: The Complete Reference, Osborne/McGraw-Hill, 2001.

Detail Description Paragraph:
[0125] In a complete DTD for the XML file of TABLE A, the data elements, such as "<LAT>38 degrees, 48 minutes North</LAT>" would be specified in the DTD as "<! ELEMENT LAT (#PCDATA)>" to indicate that these elements are present and contain only data, but do not contain not other elements.

Detail Description Paragraph:
[0133] To enable the wireless device 100 to read the XML recommendations file 250 of TABLE D, a DOM tree-based parser in the device 100 reads in the DTD of TABLE E and the XML file 250 received from the network server 140. The DOM tree-based parser converts the XML file 250 into a hierarchical tree data structure enabling the data for each element to be accessible to the application programs 106 and recommendation algorithms 112.

Detail Description Paragraph:
[0134] This process also works in reverse in the network server 140 and enables the network server to construct the XML recommendations file 250. The DOM tree-based parser can read in the DTD of TABLE E and create the hierarchical tree data structure that serves as a template for the recommendation algorithm 166 in server 140. The recommendation algorithm 166 can then fill the nodes of the tree with recommendation data, such as ADDRESS data, AREA data, LAT data, and LON data. The DOM tree-based parser uses this newly created tree of data to create a corresponding XML recommendations file 250 of TABLE D, that conforms to the DTD of TABLE E. The recommendation algorithm 166 and the DOM tree-based parser, in effect, work together as a document generator. The Document Object Model (DOM) defines the characteristics of the XML file hierarchical tree data structure and an application programming interface (API) for manipulating it. A description of DOM is provided

on the web site http://www.w3.org/TR/DOM-Level-2-Core/. For additional information on the Document Object Model and the XML file hierarchical tree data structure, see the book by Elliotte Harold, et al, entitled XML In A Nutshell, O'Reilly & Associates, 2001.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L7: Entry 5 of 7 File: PGPB Jul 11, 2002

DOCUMENT-IDENTIFIER: US 20020090934 A1
TITLE: Content and application delivery and management platform system and method

Application Filing Date:
20011120

Detail Description Paragraph:
[0089] A content delivery and management system designed in accordance with at least one embodiment of the invention may encompass future `always on` wireless infrastructure, e.g., General Packet Radio Service (GPRS) and 3G and emerging industry standards for data synchronization using Synchronized Markup Language (SyncML).

Detail Description Paragraph:
[0093] Once the field mappings have been set-up, the system packet engine(s) 710 may dynamically format data from the content provider's data source as channel data is requested by the client software included in the web management subsystem 625 running on the end user's PC and/or handheld. Tables show the format for memo (Table 1), calendar (Table 2) and contacts (Table 3) packets. Specific examples relevant to a particular content provider's data have been supplied to illustrate the best use of the data structure.

Detail Description Paragraph:
[0099] FIG. 7 shows one implementation of the web management subsystem 625 illustrated in FIG. 6. The web management subsystem 625 may include a mobile device subscription manager 720, and a desktop subscription manager 725, which may include device-based technology (for the wireless environment) and PC-based technology for the (wireline environment), respectively, that allow the end user to manage the content they are receiving. The system 625 may also include one or more data structures 730 configured to store user-specific data.

CLAIMS:

26. The system of claim 25, wherein the web management subsystem further includes at least one data structure configured to store user-specific data.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L7: Entry 6 of 7 File: PGPB Jul 4, 2002

DOCUMENT-IDENTIFIER: US 20020087596 A1
TITLE: Compact tree representation of markup languages

Abstract Paragraph:
A document written in a markup language is represented by a unique data structure. A virtual node tree describes the structure of the data types in the document. Each one of the nodes in the virtual node tree respectively corresponds to one of the data types in the document. A data array corresponding to each one of the nodes in the virtual node tree includes information identifying the relationship of the node to other nodes in the virtual node tree and a reference indicating the location of the data corresponding to the node. A set of software components obtains the data corresponding to the nodes using the references included in the data array.

Application Filing Date:
20001229

Summary of Invention Paragraph:
[0008] The lack of a practical document tree representation for mobile and other small devices has several disadvantages. For example, trees can be used to provide a common representation of documents marked up using different markup languages. For example, Document Object Model (DOM) is a standardized, widely used document tree interface for documents marked up using any of the XML-based markup languages. When DOM is available, applications which use different XML-based languages can share common software infrastructure for parsing, representing, and accessing documents. For example, in a WAP-enabled mobile phone, the WML browser, the WAP push processing software, and synchronization software that uses SyncML can all share the same parser, tree representation and document interface. This makes implementation simpler and saves memory. This is increasingly important because standards bodies, such as WAPForum and W3C, have specified a larger number of XML-based languages as standard document formats that devices must be able to handle.

Summary of Invention Paragraph:
[0010] The present invention addresses mobile devices, software applications and data structures which are disadvantageous for at least the reasons recognized above. There are several different aspects to the invention, some of which may be practiced without the others.

Summary of Invention Paragraph:
[0011] One aspect of the present invention is directed to a method of representing a document written in a markup language by a unique data structure. A virtual node tree describes the structure of the data types in the document. Each one of the nodes in the virtual node tree respectively corresponds to one element of a specific data type in the document. A data array corresponding to each one of the nodes in the virtual node tree includes information identifying the relationship of the node to other nodes in the virtual node tree and a reference indicating the location of the data corresponding to the node. A set of software components obtains the data corresponding to the nodes using the references included in the data array.

Detail Description Paragraph:
[0048] The compact tree description block is itself a minimal representation of the tree structure. Unlike a conventional tree representation where tree nodes of different types contain different data, all of the elements of the compact tree description block can be of a fixed size. This makes it possible to construct a compact tree from a document with only a single memory allocation whose size can easily be pre-calculated. This greatly simplifies memory management on a mobile device using compact trees.

Detail Description Paragraph:
[0056] A tree is a data structure composed of tree nodes. Tree nodes are data structures that contain links to other tree nodes such that the linked nodes form a branching structure like a tree. One common form of tree has a single root node which has links to child nodes; each child node has links to further child nodes; and so on to form the branching structure of the tree. Such a tree can be constructed from nodes each of which contains only two links: one to the first child and one to the next sibling. Besides links to related nodes, a tree node also must contain data elements describing the node itself. (Trees are widely used in computing, and the specific form of tree described here is well known in the art.)

Detail Description Paragraph:
[0058] In a compact tree according to the example embodiment of the invention, the tree description block contains data structures forming a tree as described above. However, the data associated with each node is represented within the node only as a field containing the location of the start of the associated data within the raw document block. So, for example, in the case of a tree constructed from an XML document, the node representing the root element contains a field giving the location of the beginning of the root element in the raw XML document block.

Detail Description Paragraph:
[0070] However, the types of elements stored in a document typically do not correspond to the storage types which are operated on by programs. Programs operate on primitive data types of various lengths, such as chars, integers, pointers and structured data elements built from these primitive types. Moreover, the same types may be stored differently in different computers. For example, integers may be stored with different byte orders. Because of these differences, it is generally not possible to treat a data element contained in a raw document as if it were one of the internal types used by a program. For example, if a program has the address in memory of some element of a raw document, it cannot typically treat this as the address of some structured data type understood by the program. Instead, programs which wish to operate on raw document data need some component to convert back and forth (serialize and deserialize) the data between raw document storage data and structured, typed, data. Since compact document trees use the raw document to store all of the document data, reading or writing any data element from the tree requires deserialization or serialization of the raw document data.

Detail Description Paragraph:
[0084] A set of type serializers is instantiated, only one of which is shown in FIG. 9 and each of which is capable of writing the raw data corresponding to some data element. These are preferably implemented as a set of functions that can be called with parameters that describe the raw data and the location to which it is to be written.

Detail Description Paragraph:
[0091] Although a tree written by this method can be read exacty like a read-only compact tree, it may not be as space-efficient for two reasons. Since it may be necessary to add some context to raw data elements so that they can be deserialized correctly, raw elements may be larger. However, any additional space required should typically be quite small. For example, with WBXML, the only additional raw data required is to encode code pages, and most WBXML documents contain little or

no code page data. Depending on the allocation mechanism for new blocks, some allocation overhead (such as block headers) may also be required.

Detail Description Paragraph:
[0095] Compact trees may be used either as an internal representation of tree structure used by computational algorithms that require tree structures or as a method for storing and exchanging tree-structured data. When used as an internal representation, compact trees are implemented in whatever way is most suitable to the computational algorithm using them. For example, the tree description block may be implemented as an array stored in memory, using whatever data structures are required by the algorithm and the computing system and language used to implement the algorithm. Such a representation is internal in the sense that it is used by the computational algorithm in ways and in a form that may not be known outside the algorithm.

Detail Description Paragraph:
[0098] A number of additional uses of the structure of compact trees and methods for reading and writing node data are possible. A specific serialization format may serialize compact trees with variable length tree nodes as multi-part documents. Also, a compact tree encoder may implement a memory-efficient component for generating a raw document block from a read/write compact tree. A raw document is an encoding of data with a particular format. For example, XML documents can be encoded either as simple text or in binary format (WBXML). Using the compact tree encoder, a device can construct a compact tree and then output an encoded document from the constructed tree for storage or exchange with other devices. An example would be synchronization software in one device which constructs a Synchronization Markup Language (SyncML) compact tree and then outputs a WBXML-encoded SyncML document to another device in order to accomplish synchronization.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L9: Entry 2 of 3 File: USPT Sep 23, 2003

DOCUMENT-IDENTIFIER: US 6625621 B2
TITLE: System and methods for a fast and scalable synchronization server

Application Filing Date (1):
20001214

Drawing Description Text (10):
FIG. 7A is a table illustrating messages that may be sent between a sync client and a sync server when the sync client is sending changes to the sync server during a synchronization process.

Drawing Description Text (11):
FIG. 7B is a table illustrating messages that may be sent between a sync client and a sync server when the sync server is sending changes to the sync client during a synchronization process.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L9: Entry 1 of 3 File: PGPB Jun 20, 2002

DOCUMENT-IDENTIFIER: US 20020078072 A1
TITLE: System and methods for a fast and scalable synchronization server

Application Filing Date:
20001214

Brief Description of Drawings Paragraph:
[0020] FIG. 7A is a table illustrating messages that may be sent between a sync client and a sync server when the sync client is sending changes to the sync server during a synchronization process.

Brief Description of Drawings Paragraph:
[0021] FIG. 7B is a table illustrating messages that may be sent between a sync client and a sync server when the sync server is sending changes to the sync client during a synchronization process.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit

Generate Collection

L7: Entry 4 of 7 File: PGPB Nov 28, 2002

DOCUMENT-IDENTIFIER: US 20020178211 A1
TITLE: Technique for enabling remote data access and manipulation from a pervasive device

Application Filing Date:
20010503

Detail Description Paragraph:
[0036] Alternative embodiments may include different and/or additional protocol proxy types. For example, a synchronization protocol proxy may be included, which may be used to synchronize data stored locally on a user's WID with data stored elsewhere (such as on the user's desktop PC). An example synchronization protocol is "SyncML" which is being developed by The SyncML Initiative to seamlessly synchronize wireless and wireline data and devices. (See http://www.syncml.org for more information on SyncML.)

CLAIMS:

35. The method according to claim 1, wherein the determining step further comprises accessing a data structure to locate information used by the returning step, wherein the data structure stores information about the data manipulation operations that are available for the obtained data and the location of each available data manipulation operation.

36. The method according to claim 35, wherein new data manipulation operations are supported for use in the determining step by adding information about the new data manipulation operations and the location of each new data manipulation operation to the data structure.

37. The method according to claim 1, wherein the determining step further comprises: accessing a data structure to locate information used by the returning step, wherein the data structure stores information about the data manipulation operations that are available for the obtained data; and dynamically determining the location of each available data manipulation operation.

Previous Doc Next Doc Go to Doc#